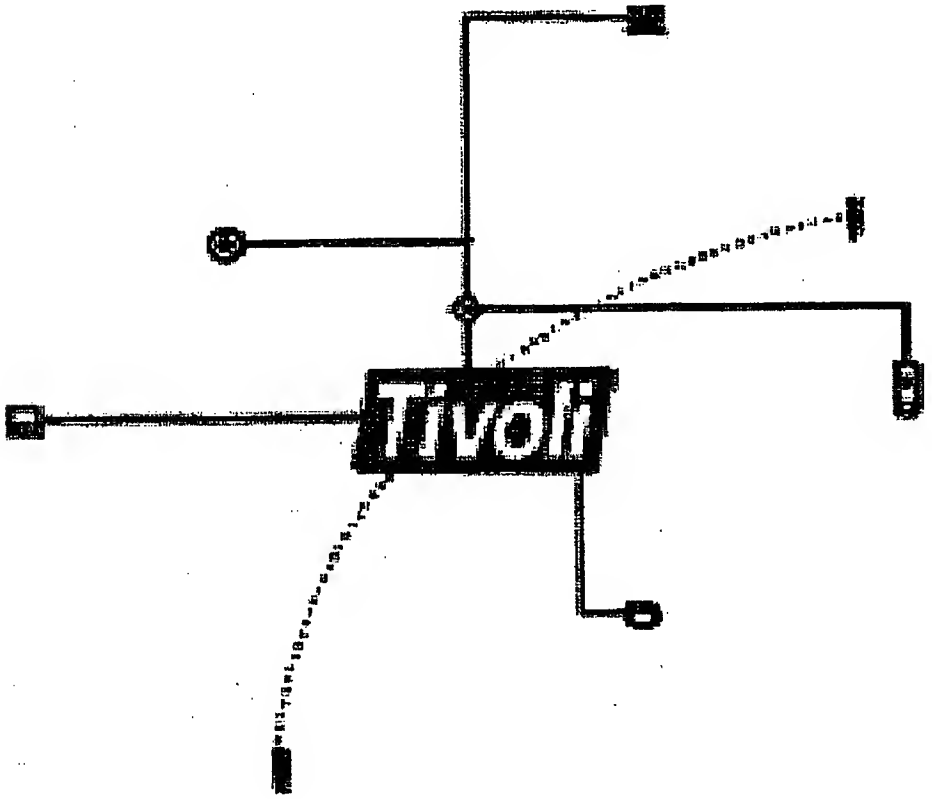
Exhibit "A"

Tivoli Technology Strategy

"Leverage for Speed"



Version: Draft 0.2

Last Modified: [redacted]

Tivoli Confidential

Table of Contents

Abstract

The Tivoli Technology strategy is the instrument that translates the Tivoli corporate strategy into meaningful development methodologies, and provides the framework for technical decision making. To achieve this it must be set in the context of Tivoli's current and future business environment.

Executive Summary

The business environment is assumed to be growth oriented and dynamic. The business will expand through new market initiative, and through acquisition. These new initiatives will still require resource to be managed, although the definition of resource may be much broader than it is today. The approach or philosophy used to manage them may very greatly.

Because of this it will never be possible, and indeed is not desirable to converge on a single technological solution or architecture. The technology infrastructure must be able to easily accommodate infusions of new technology, new kinds of resources, and new approaches to managing them. The technical strategy must enable Tivoli to easily support new market initiatives by creating a flexible technology base.

To accomplish this, the technical strategy is based on management topographies. A topography is the codification of the requirements of a specific management solution. It is assumed that Tivoli will require many topographies to support the diverse market initiatives. For Tivoli to be successful, the development community must be able to easily create and maintain a diverse set of topographies.

This is accomplished by a component architecture. Topographies are composed of components. The components are organized into categories that are archetypal management services. Because both the function of component categories and the boundaries between them are well understood, it becomes comparatively easy to recombine components to form new topographies, and easy to determine when new component instances are required. A single topography may be composed of components with widely varying implementations.

The component approach focuses on specific types of management services as opposed to management philosophies that are emphasized by a framework centric approach. This allows competencies to be developed around various types of management service that may be leveraged by many market initiatives.

Purpose

[This section is to describe the audience and how they are to use this document. Senior management and BU leadership]

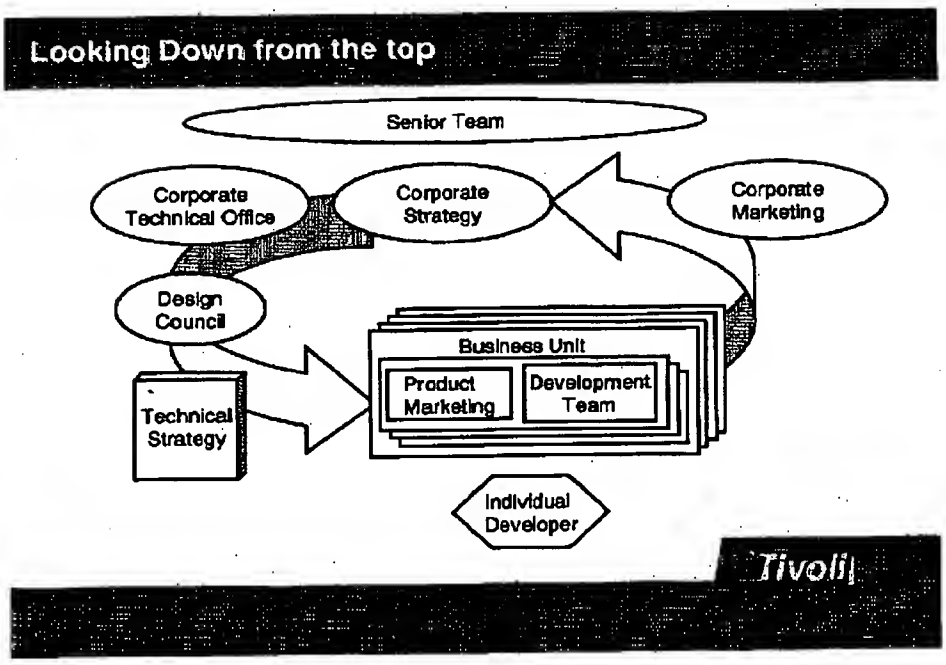
The purpose of the technical strategy is to provide the leaders of the market initiative units (sometimes called Tivoli Business Units) the materials they need to understand what is and is not corporate technical strategy decisions.

The technical strategy must facilitate a "leverage for speed" engineering culture within Tivoli.

It is important to note that this document is not intended for the general population at Tivoli. It is the responsibilities of the unit leaders to translate the relevant content of this strategy into material that make senses for their teams.

Introduction

[This section needs to explain the following chart...]



The Tivoli Technology strategy is the instrument that translates the Tivoli corporate strategy into meaningful development methodologies, and provides the

framework for technical decision making. To achieve this is must be set in the context of Tivoli's current and future business environment.

It describes the operating environment that allows business units to define development plans for their products in the context of the Corporate Architecture.

The corporate strategy calls for continuing investment in new and emerging markets. To support these initiatives, the development community will be continual be asked to quickly provide new offerings. To accomplish this, the new offering must be able to easily leverage existing technology assets.

Some high level questions...

1. *How can technology be harnessed to meet the needs of our existing markets?*
2. *How can technology be exploited to create new business opportunities?*
3. *How can technology be managed to achieve economy of scale?*

What type of business is Tivoli?

[By explaining how we have changed – setup why we need something new. Might not hurt to lay out that we are one of the 10 biggest software companies so what works for the smaller guys may not work for us]

Tivoli System's business is a ***profitable growth*** business that is incorporating more products and is expanding into new and changing markets at a demanding rate. The technical strategy for Tivoli must systematically address this challenging environment by facilitating speed.

This means Tivoli continues to be a company in transition. Tivoli's first major transition was from a distributed object infrastructure provider to a management application provider. Then, Tivoli shifted from a single geography company to a global software vendor through the IBM acquisition. Then, Tivoli transitioned from a single business company into a multiple business company to position itself for growth. The one pattern that remains constant in these transitions is the subsequent one is more challenging and more difficult then the one before.

Tivoli's Mission, Vision, and Values

[Build a short discussion around this material that leads into the role of the technical strategy....]

Mission: To be the driving force in the changing role of technology by providing management solutions that allow our customers to unlock the power of technology.

Visions Goals:

(1) $25 Billion by 2005

(2) We are Number 1 in Customer Loyality

(3) We are the Most Sought After Place to Work

(4) Manage Anything Anywhere 1 Billion Devices by 2005

Visions Description:

(add them here)

Values:

(add them here)

Tivoli's Corporate Strategy

[This section is to describe the aspects of the corporate strategy that sets the context for the technical strategy: diverse overlapping markets]

Multiple markets is the next major transition in the business

I. Business Model Methodology

II. Nerve's Wire's Future Mapping Methodology

In order to provide some order to the chaos surrounding the changing market place, Tivoli is making strategic investment decisions using an "end-state methodology". End states are outcomes that describe the enterprise IT environment and the management software industry over the strategic planning horizon. Based on this view, investments are balance based on the perceive likelihood of the end state and the benefit to Tivoli. The current end states are covered in the following table[1].

End State	Main Themes
A: It's All About Managing Services	• Enterprises purchase a wide variety of IT services from diverse service providers. • Product sales turn into service sales. • Service and security management are the prime concern of both internal and external service providers. • Management is available as a service offering to the enterprise.
B: Providing Higher Value-Added Management	• E-Business and the Internet create new management challenges and opportunities • Management vendors provide new value-added point product and focused suite solutions to managing digital assets, network-based data warehousing, multimedia communications, product telemetry, converged network services, etc. • Equipment makers address traditional network and system management problems.

[1] Note that the end states are not distinct. This is by design to reflect the reality that many of these end-states reflect competing and/or contrasting approaches that will co-exist in the market place.

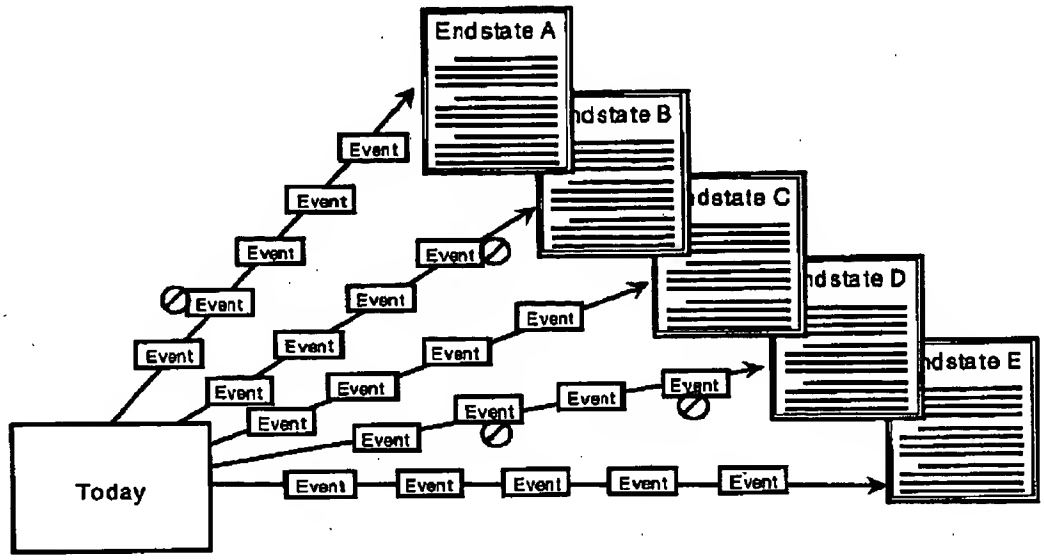
C: Managing Virtual Value Webs and Net Markets	• Enterprises participate in an increasingly complex web of electronic relationships and net markets; many adopt a more virtual approach to business. • Detailed visibility into electronic links with market hubs, customers, suppliers, and partners is critical to success. • Companies look to management vendors to assist them in creating new efficiencies and competitive differentiators, and to optimize their e-business links, transactions, and information flows. • Management vendors have also adopted the ecosystem model.
D: Get Real	• The G2000 enterprises change slowly and still struggle with the traditional problems and disciplines of network, system, and application management. • Enterprises value management software that copes with both legacy systems and new technology. • Nothing ever seems to go away – it just sinks deeper under layers and layers of new systems and software.
E: Ubiquitous, Built-In Management of the Global Utility	• A robust, global computing utility infrastructure supports a proliferation of continuously connected devices, information kiosks and smart, connected industrial systems (e.g., networked gas pumps) for consumers and business computing markets. • Management vendors provide SIs, system vendors, and large enterprises with componentized offerings designed for integration with other products and custom software. • Always on and always available broadband, wireless access is prevalent to the home and businesses. The rapid rise of the mobile Internet, which started outside of the United States, has created a whole new generation of services, products, and portals.

Because both the end state methodology is a predictive process, both the set of end states, and their likelihood varies across planning cycles. The emergence of new end-states may create the need to invest in new and previous unanticipated areas.

What is a Business Unit? Multiple, overlapping marketing initiatives

[Describe the fact that there are multiple marketing initiatives, we will be creating more, our long term viability depends on our ability to play in multiple and to capture sales when organization go through transitions.]

(So, describe the following)



Standard Operating Environment

[This section need to expose the realities of Tivoli's environments that prevent goals like a single source tree across the company.]

Tivoli's corporate strategy makes is necessary to assume standard Tivoli operating environment is characterized by the following:

1. There will be overlap in the functionality of our products because it will be necessary to experiment in new markets by leveraging what we have and what we can quickly bring into our portfolio.
2. The rate of change in both the markets and technologies will make it impossible for Tivoli to converge on a single technology base.
3. New technologies will be incorporated into the mix through acquisitions or similar activities on a regular basis
4. Multiple markets initiatives will impose conflicting demands upon solutions. (Multiple packaging, licensing, and pricing schemes.)
5. The strategy must be executable in the context of divergent end states. For example, we will ever have a single framework.

Technical Strategy Criteria

[This section needs to set up the criteria from the point of view of what would be true if the technical strategy match the corporate strategy]

Therefore, a successful technical strategy must meet the following criteria:

1. The technical architecture and the product packaging strategy must be independent. It must be assumed that Tivoli technologies will be packaged and delivered into different markets at different times.

   Key Question: How many different initiatives can we effectively handle at a time?

2. It must provide economy of scale by enabling a wide reuse of technical expertise and implementations.
3. It must easily accommodate infusion of new technology without forcing major rewrites to conform to a master blueprint.
4. It must create a context for independent decision-making that does not interfere with Tivoli's business objectives.
5. It must be durable through organizational changes....restructuring of business units does not signal a change in Tivoli's technical strategy.

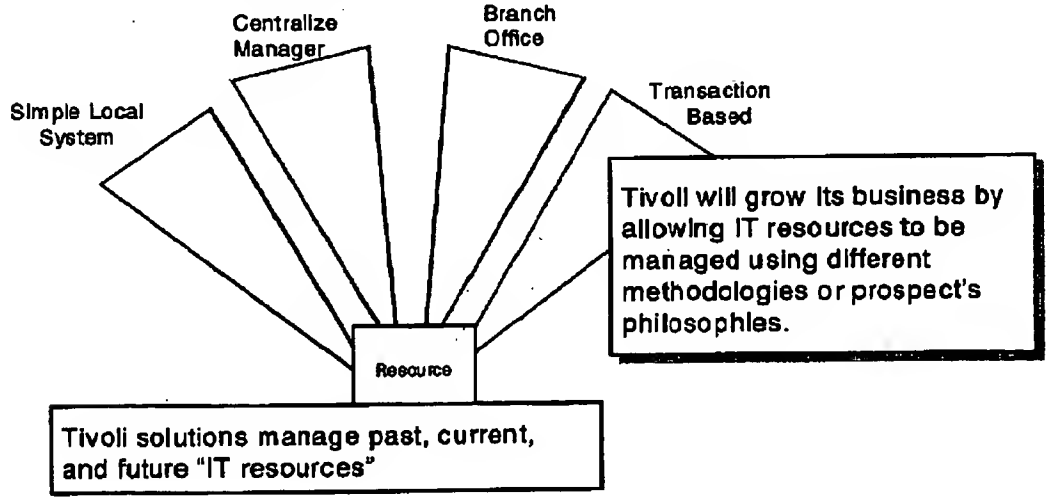
Where does the Tivoli Technical Strategy Begin?

[What is constant about out business – Assumptions we've made]

With all this change and overlapping views, the technical strategy needs a context that is durable over the strategic planning horizon. This context is based on the following two assertions:

1. Tivoli solutions manage past, current and future "IT resources". Future "IT" resources include notions like appliances, mobile devices, etc.
2. Tivoli will grow its business by allowing IT resources to be managed using different methodologies or management philosophies.

The management philosophy notion is an essential variant because one could simply describe Tivoli's growth strategy by asserting it is trying to accommodate multiple philosophies of managing resource.



This notion can be illustrated by looking at each of the end states. A short, and incomplete, view of the management philosophy for each of the end states is:

(1) End State A – It is about managing services provided to many different companies with a control mentality

(2) End State B – it is about managing higher order notions like business processes.

(3) End State C – management becomes part of a technologies that facilitate the formation of virtual communities

(4) End State D – IT organization continue to manage the heterogeneity

(5) End State E – Management is no longer an issue.

The execution of the technical strategy must systematically lead to more flexibility[2] in the technologies Tivoli uses so Tivoli can create new business opportunities.

This technology strategy seeks to leverage our competencies in providing these services rather than depend on any specific technologies or infrastructures.

How is a market initiative different from a startup company?

[Discuss why we can't run new market initiative like start-up – Global reputation raises the bar – Reliability, Scalability, I18N]

What demands does the corporate strategy place on technology?

[Describe the things that our technology/architecture/development process must be able to accomplish for the corporate strategy to succeed]

Translate the criteria for success into technology and architecture issues

Tivoli's technical strategy identifies the important interfaces by defining components.

Tivoli's technical strategy is based on a "component" architecture. "Component" approaches require a common understanding of the scheme or taxonomy used to characterize good or well-behaved components. Good or well-behaved

[2] *How do we measure flexibility?*

components within the Tivoli architecture are those that can be recombined to implement different management philosophies for a set of resources. A combination of components that implement a particular management philosophy is referred to as topography.

Tivoli's technology strategy seeks to increase flexibility by segregating variation, and by defining clear boundaries between components of the management structure that vary along different axis's.

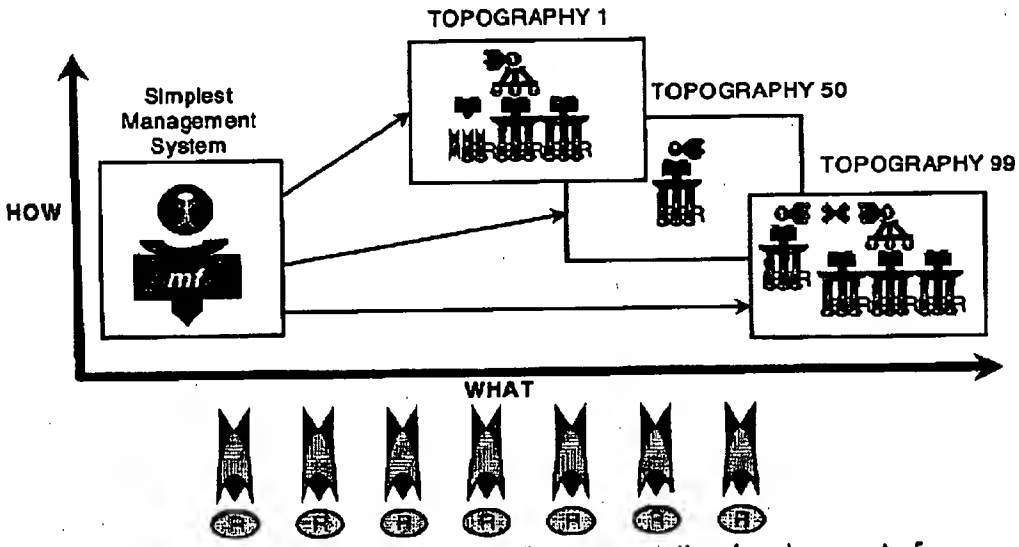
This technical strategy allow Tivoli to vary the following parameters to expand its business:

(1) What we manage...

(2) How we manage....

(3) How we deliver it...

Tivoli's value-add increase with both the number of resources managed and the complexity of the management philosophy.



A major leg of our technical strategy is to "encourage the development of management philosophy neutral components with the intent that they will be deployed and package for many topographies".

Implications of Corporate strategy

[This Section explains the specific requirements placed on the technical strategy by corporate strategy. That is the requirements for flexibility and dynamic technology leverage – Tie to operating environment]

Explain why we can't converge on single framework, programming model, source tree etc.

What is the right way to do system management?

[This section introduces management philosophy and explains why it is the fundamental variant in what technology is required to support a market initiative – Point 2 from context]

[Explain shift form Tivoli advocating a management philosophy to seeking to support customers philosophy]

Prospects that intend to use Tivoli solutions are predisposed to the "right" way to attack management. This predisposition is a blending of beliefs that address the following areas:

- Who performs what management activities?
- How are these folks organized? That is, what is the correct structure of the IT organization?
- Where are decisions about management activities made?
- Who manages what resources?
- How is management capability deployed into the environment?
- How much influence does the IT organization have over the consumers of the IT resources?
- What IT resource are managed?

This predisposition can be characterized as a management philosophy. A management philosophy expresses a set of beliefs or predisposition's about the right or desirable way to approach management. These beliefs go beyond technology or the mechanics of a specific management product.

Management products tend to codify in the product architecture one, or a set of related, management philosophies. These philosophies proscribe the set of markets the product can address.

The technical strategy seeks to limit the expression of a Management philosophy to a well-defined set of components, and where possible, to have a component only express a single facet of the philosophy. In addition, some aspects of the philosophy can be expressed in the arrangement of components so they need not be part of any component.

Growth can not be sustained by limiting Tivoli's solutions to a single management philosophy or methodology. In the same way that Tivoli cannot limit its language support: If Tivoli limits itself, it is choosing to ignore a part of the market. This requires that the technology be positioned to support a wide range of management philosophies, and to easily accommodate new philosophies as they emerge.

It was fine to advocate a management philosophy when the large portion of the internal enterprise market that accept it were sufficient to drive our growth. Supporting multiple management philosophies is market expanding just like internationalizing our products was.

Anything is a lot

[This section discusses the explosive growth in number of kinds of things we have to manage that is implied by the corporate strategy – Pont 1 from context]

So where is Anywhere?

[Discuss the need to be able to quickly produce management solutions that work in new environments]

Both the components used by a topography and their characteristics are impacted by the hosting environment of the components and the physical deployment environment or distributed topology of a management solution. In addition, management solution must adapt to both the organizational structures of the managed environment, and of the operating organization. Some aspects of deployment reflect management philosophy and some reflect physical constraints.

How can Tivoli leverage its successes?

In a framework approach, applications and resource interfaces are tied to the framework. A new framework means new everything. In a component approach, only the things that need to be different are rebuild

Frameworks X Applications X Resource types = Development effort

Components + Backplane + Resource types = Development effort

Components – Designing for speed

This section describes what Tivoli's Component Architecture is, why is it important to the strategy, and references the component architecture document:

The strategy addresses its requirements by delineating well defined boundaries between components of the technology portfolio to allow each component to vary independently of others, and to reduce the amount of time it takes to assemble components for different market initiatives.

How is a management solution defined?

[This section describes how the concept of topography can be used to codify and graphically depict the technology and architecture required to support a market initiative Criteria 2]

The definition of "topography" is the art or practice of graphic delineation that shows the relative position of important surfaces.

In the context of Tivoli's Technical Strategy, the connotation of "topography" is the art or practice of delineated the relative position of components that combine to embody a particular management philosophy. Therefore, a solution that embodies a particular management philosophy is a particular topography. When an engineer become familiar with a particular topography they ought to understand how a family of components can be used to deliver a particular solution.

Topography provides a common vocabulary for describing discussing and analyzing the solutions required by various parts of Tivoli's business. This vocabulary is not colored by any implementation approach or programming model.

The goal of topographies is to isolate the management philosophy dependencies in a well-defined and understood set of components that compose the management infrastructure.

The premise behind this is: there is a relatively standard set of management services or activities that are common to most market initiates, but the way they

are deployed and used varies greatly. This implies that leverage may be obtained by separating the management activities from their employment.

Management Topography Example

The topography notion can be used to illustrated the differences between the following two management philosophies that are common in distributed management environments:

(1) Management functions should be located on a centralized server with feeders to the resources to minimize the resource consumed on the end points and to allow more cross resource decisions making.

(2) Management services ought to be close to the resource so key decision can be made locally and so the decision making capability is not exposed to network outages.

This can be illustrated by contrasting two approaches to Implementing monitoring.

The topographical components (see "Component Overview" on page 23 for symbol legend) that will be used in this example are:

(1) Activity

(2) Management Actor - The monitoring function is configured to retrieve specific attributes from a selected set of resources at predefined time intervals. The monitor feeds the values to an analysis engine that performs threshold type analysis on the raw data, and cross resource correlation.

(3) Touch Point – This component knows how to interact with a particular type of resource.

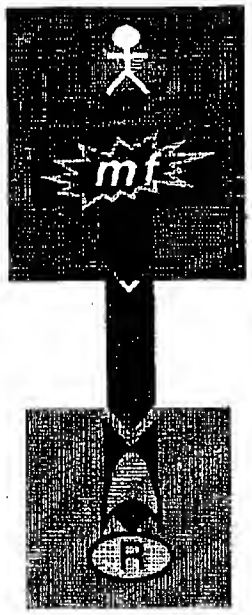
(4) Management Operator Conduit – This is a component that transports a management operation from one machine to another.

(5) Management Activity Conduit. This is a component that transports a management activity from one machine to another.

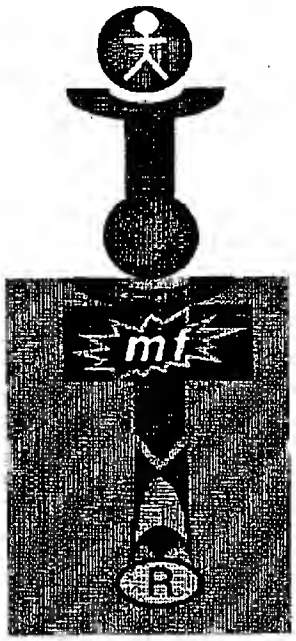
The following graphic illustrates how the two different management philosophies can be implemented use common topographical components.

Management Function Remotely Accesses Resource

Remote Invocation of Local Management



Management Function Remotely Accesses Resource



Remote Invocation of Local Management

The management philosophy is captured in two ways:

1. The arrangement and selection of the topographical component.
2. The characteristics of the individual components used to build a topography

Topographical Components

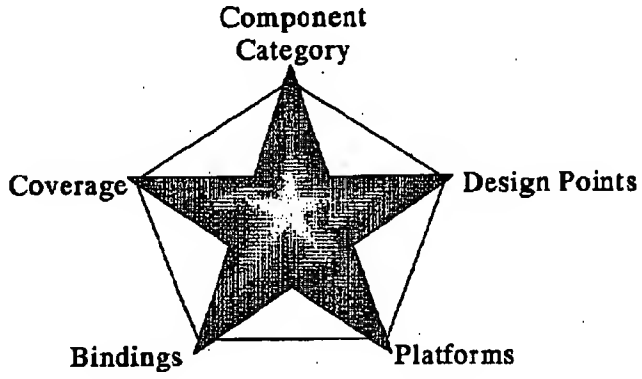
A topographical component is a component category that clusters a set of design discussions. Each component category clusters a set of abstract services. They are abstract because an implementation and programming model are not specified. The topographical components form a taxonomy that allows engineers to categorize the capabilities of a particular implementation.

A *component category* clusters a set of design discussions. The type of information captured for a particular category is its *description*, its *coverage dimension*, and the *Design Points* for each. The description is simply that: an informative description. The coverage dimension describes how Tivoli decides that it needs another component in the category. The design points are the factors that describe the specific capability of an instance in this category.

An example of a component category is "touch points". This category is a direct derivative of one of the two assertions the drives Tivoli's technical strategy: Our solutions manage past, current and future "IT resources".

How many components do we need?

[Set up design points and component categories in a non technical way – We still build more than one of the same type of thing, but we build multiples because it needs to function differently not because it has to work in a new configuration]



Platform Coverage (Porting)

Topographical Backplane

The purpose of the backplane is to provide an environment in which components can easily be configured in many different configurations. The backplane can be thought of as the surface on which the components of a topography are assembled. It should not be thought of as an object framework in which components that conform to its requirements are instantly usable. It is closer to an enterprise application integration tool in which a modest amount of custom work is required to integrate each component.

To accomplish this, a backplane needs to provide the following:

- A standard definition of how components interact. It may, but need not provide support for the interactions. This is what allows components to be easily reconfigured.
- Services that must be available for all components. Services that are provided by the backplane should be very management philosophy neutral. Ubiquitous industry standards are good candidates. Two possible candidates for backplane services are authentication/authorization, and deployment/installation

How are components built?

[Describe role of programming libraries and support services such as DAS]

How is a topography different from a framework

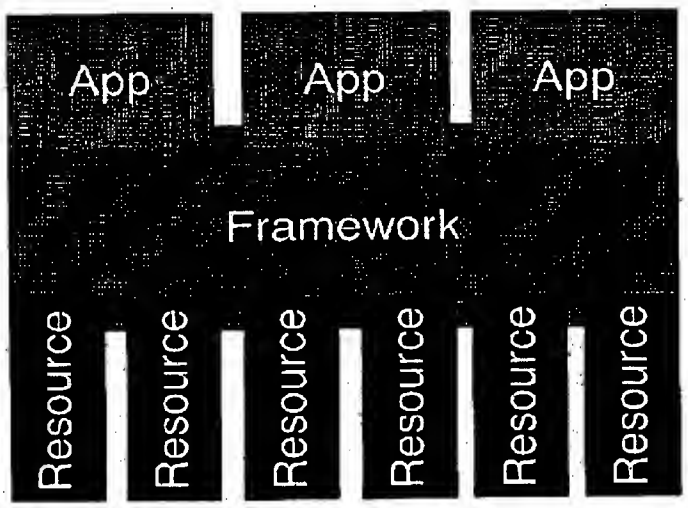
The components of a management system can be loosely organized into two categories:

- Those that support specific management services
- Those that provide infrastructure that is usable by many management services.

For a given implementation of a management system, the infrastructure is common supporting many management services. Because of how it is implemented, the infrastructure is often referred to as a framework, and the various management services as application. The framework embodies a philosophy or approach to management. The applications are relatively neutral, but tend to reflect the approach of the hosting framework.

A framework is intended to provide a common set of functions that are reusable by all of the management services it supports. Frameworks provide solutions to problems that are common to all management services such as distributed communication authorization, and resource selection.

A framework based management system might be depicted as:



The framework defines characteristics of the management system such as the:

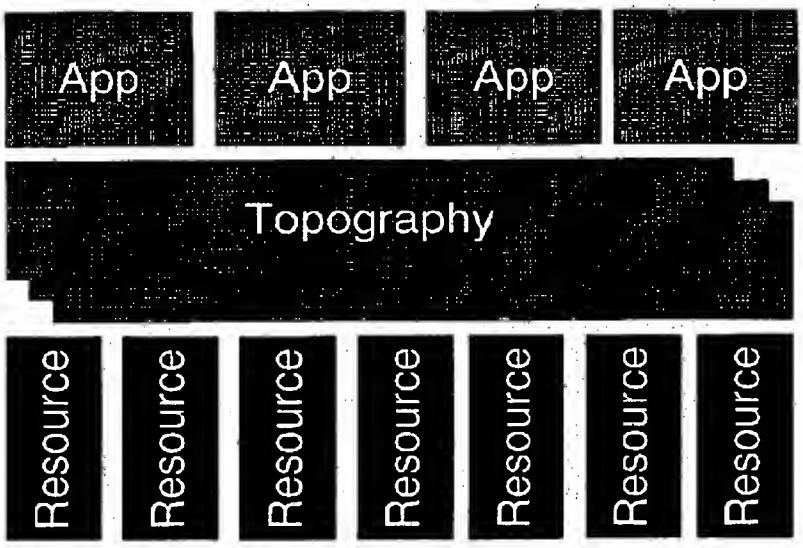
- Control model
- Scale
- Resource agragation mechanism
- Distrubuted topology

- Resource interaction style

Collectively, these define the management philosophy the supported deployment environment and the supported resource mix.

The framework model works well when it is only necessary to support a single management philosophy. However, when a framework is stretched to implement multiple philosophies that are not closely related it becomes awkward and inconsistent.

Because most of the characteristics of a management philosophy are expressed in common infrastructure, it is possible to build management services, and resource interfaces that are philosophy neutral. When the characteristics of a management service that are effected by philosophy are well delineated from those that are not, it is possible to build management services that can support multiple philosophies. This approach can be depicted as:



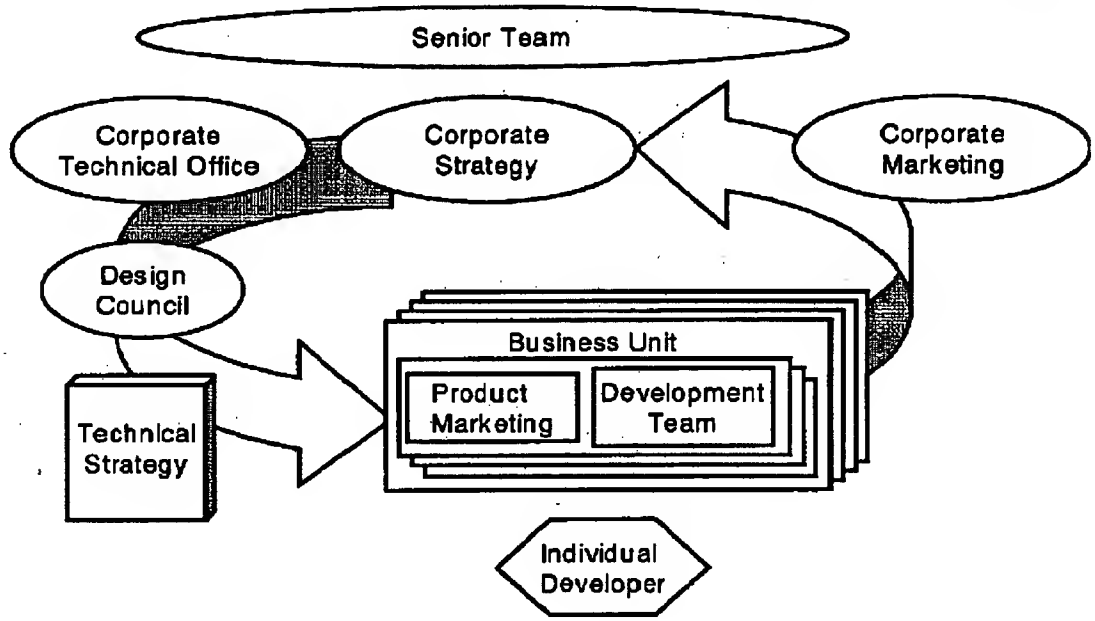
Topographies differ from frameworks in that frameworks explicitly publish a management philosophy that is expected to permeate services that are implemented on top of the framework. Topographies explicitly localize facets of management philosophy into well-defined components. The philosophy supported by a topography based management system can be changed by replacing or rearranging those components that express it. Most components, especially those that implement "management applications" have any expression of management philosophy distilled out of them.

How does a topography become a product?

[Discuss what needs to be done to assemble components into products – Make clear that there is some work – its not magic – Criteria 1]

Creating a culture for speed

[This section outlines how the technical strategy is put into operation – That is it lays the foundation for a detailed operation plan]



How are technical decisions made?

[This section outlines who is responsible for what kind of decisions Criteria 4]

Role of CTO's office

[Describe the role of the CTO's office as keeper and standard bearer of a unified technical and architectural vision]

Role of Design Council

[Describe design councils role as a bridge between the CTO's office and the development community]

Role of Business units

[Describe the role of business unit leadership in projecting the requirements of the technical strategy into individual product plans]

Empowering the development community

[Describe how the strategy creates a structure that allows technical decisions to be federated]

For decisions that relate to well understood component categories: Decisions can be delegated to a person/group responsible for the component category. (This meets goal 4). Because the component categories are inherently cross topography (in today's structure – cross framework), there is an inclination to making the decisions in a strategic context.

What does the technical ownership of a component category look like?

Should there be a Product Manager for each component category?

How are new component categories created?

What about the backplane?

How are TST type projects handles? I think TMD becomes part of the backplane. What about DAS?

Operating Environment

How is the technical strategy enforced?

How is the technical strategy enhanced?

How are exceptions granted?

Approach to development

[Explain how structuring most of the development process around component categories creates stability in the swirl of varying market initiatives. Criteria 5]

Creating a culture for speed

[Describe the cultural context for development that complements the technical strategy]

There are several fundamental shifts implied by the technology strategy

It is ok to have more than one technology. The competition to determine what will be "the next framework" is meaningless

We ask our customers what management philosophy they want instead of telling them what we have.

We seek to enable variation instead of striving for uniformity

Component Categories reflect competencies

A key aspect of Tivoli's technical strategy is a shift in emphasis from a framework to a collection of topographical components that reflect a set of competencies and support technologies that allow us to address prospects with different management philosophies.

Integrating New Technologies into the Tivoli Fabric

[This section should accomplish two objectives. First, it should describe the systematic steps Tivoli ought to use to map acquired technology into the Tivoli family. Second, these same steps ought to be used to map our current technologies into this approach.]

New Initiatives

In order to facilitate leverage, we move into new market initiatives by asking two questions:

(1) which of the existing topographies is close?

(2) What needs to change to address the new initiative?

There are several flavors of change.....new backplane, change design points,

Research

Lay out criteria for evaluating research proposal

Describe how successful research projects are moved into production

Planned investigation on new technology.

What about unplanned investigation that occurs as a result of the normal development process?

Acquisitions

[Criteria 3]

How do we evaluate the viability of an acquisition?

What is the road map for integrating newly acquired technology?

What steps must an acquired product take to be a full member of the Tivoli product family?

Existing products

[Based on the acquisition road map – develop a road map for rolling out the strategy into the organization – This will eventual be its own document]

We need to start factoring out framework dependencies from applications and touch-points.

Moving on to the new

[This section defines the strategy for maturing and obsoleting technology]

Standards

Summary

Grid of criteria and the solution

Appendix - Topography Technical Overview

All topographies are composed of a set of standard components. A specific topography is defined by a combination of the way the components are assembled and the characteristics or design points defining each component.

For example, a standard component is a management operation conduit. The conduit carries management operation between a management function and a resource. In a particular topography, the conduit may be implemented as a synchronous CORBA connection. In a second topography, the conduit may be implemented as an asynchronous message based connection. In a third (simpler) topography, the management function may be directly connected to the resource so the conduit is not present. The implementor of the management function need not be aware of the characteristics of the conduit.

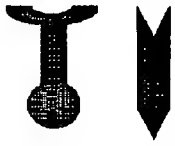
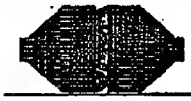
The components define the major characteristics of a management system. These include:

- How management activity is initiated
- Where management activity is approved or authorized
- How management activity is communicated between the management console and the resource.
- Where specific management services interact with the management infrastructure.

The structure of a topography can be represented graphically. The organization of the components expresses the structure of a management solution. The infrastructure components are represented by standard set of symbols that can be used to build topography diagrams. Symbols for the specific management functions that comprise a management service can be developed and used to depict how these functions interact with a specific infrastructure model.

Component Overview

A sample set of topographic components is shown in the following table. The set is not exhaustive; it represents a relatively well understood set of components for common infrastructure that are found in most topographies.

Symbol	Category
	**Management Activities**
	**Resource**
	**Resource "touch-point"**
	**Management functions**
	**Conduits**
	**Multiplexers**
	**Inter-management system bridge**

Design points

The topographical components are an archetype or pattern for a specific kind or function. Design points describe the characteristics of a specific instance of a component. Each design point describes a fundamental characteristic of one or more components. Each design point has an associated set of values it may have. In most cases, the values are a discrete set not a range.

The values discussed for each design point are the know set of values based on analysis of existing applications. They should not be taken as a complete set.

As the implementation of the technical strategy matures, many of these design points will be associated with common programming or implementation services. When such services exist, they could be used to implement a design point specified for a particular topography component.

Design Point Example - Processing Style

On demand

This type of touch point goes into session with the resource when a operation is requested so all operations are wrapped with a close and an open.

Session Based

This type of touch-point hold an open session and waits for operation requests like get or set configuration setting.

Staged

This type of touch points caches the current values of the resource at some interval and the operations work against the cached values.

Topographical Backplane

Security

Installation

There is a widely accepted view among customer about how installation should be performed. It can be paraphrased as "I want it to happen by magic, and the result to reflect my preferences". Since all components require installation, and the magic can be the same for all components as long as it works for all topographies, it is a good candidates for the backplane.

While it is possible to perform topographical analysis with an abstract back plan, it is not possible to define meaningful implementations of components. A concrete backplane imposes constraints on all components that are to be used with it. These constraints take two forms:

- The character of, and interface with the services actually provided by the backplane
- The inter-component interfaces which are specified by the backplane.

Concrete backplanes may not be able to provide complete flexibility because of issues of scale and footprint. In fact, a basic premise of this strategy is that many backplanes will exist and that this is OK.

Because the backplane has some of the characteristics of a framework, it would be easy to think of it as such. It differs from a framework in several important ways. They are:

- Although it defines the semantics of component interaction, it seeks to do so in as programming model neutral a way as possible

- It seeks to be a thin as possible. Many of the services of a traditional framework are provided by interchangeable components
- Although it may provide some communication services, it is not the definitive means of communicating within a management solution. Conduit components may implement communication service where appropriate.

Deployment

Deployment reflects the way components of a management solution are placed in the physical environment as defined by both where services are hosted and virtual distance as measured by cost and availability of bandwidth.

Pre Loaded or Static

First Used

On Demand

Transation based